Wax

Packet Radio Temporary Note # 105

NIC # 30788

Packet Radio Station Design Specification #1

May 1, 1974

Dave Brandin, Stanford Research Institute

Jerry Burchfiel, Bolt Beranek and Newman

Ray Tomlinson, Bolt Beranek and Newman

Dale Walls, Collins Radio, Inc.

I. Introduction

This note provides a preliminary design for the Packet Radio station. Issues addressed here include equipment configuration, interface standardization, and partition of functional responsibility among equipment modules. Issues not addressed here include routing and repeater labelling algorithms and protocols, station-terminal protocols, and ARPANET-PRN gateway algorithms. These protocol issues will be discussed in later documents.

II. Equipment Configuration

Two goals were foremost in specification of the station equipment: first, the standardization of equipment modules to permit easy stocking of spares and interchangeability, as well as minimum module count. This goal resulted in selection of common hardware for the terminal, repeater, and station; all will use the same RF module, signal processor, digital interface, and microprocessor. The second goal was interface standardization to provide standard means for equipment interconnection. A separate document, (Specification of a Standard Packet Interface, Packet Radio Temporary Note # 10) details this specification. This standard interface permits simple self-looping and cross-coupling experiments; it is used for the interconnection of the modules described below.

The station consists of the three modules shown in Figure 1: a Station Processor, a Repeater Processor, and a Repeater Transceiver.

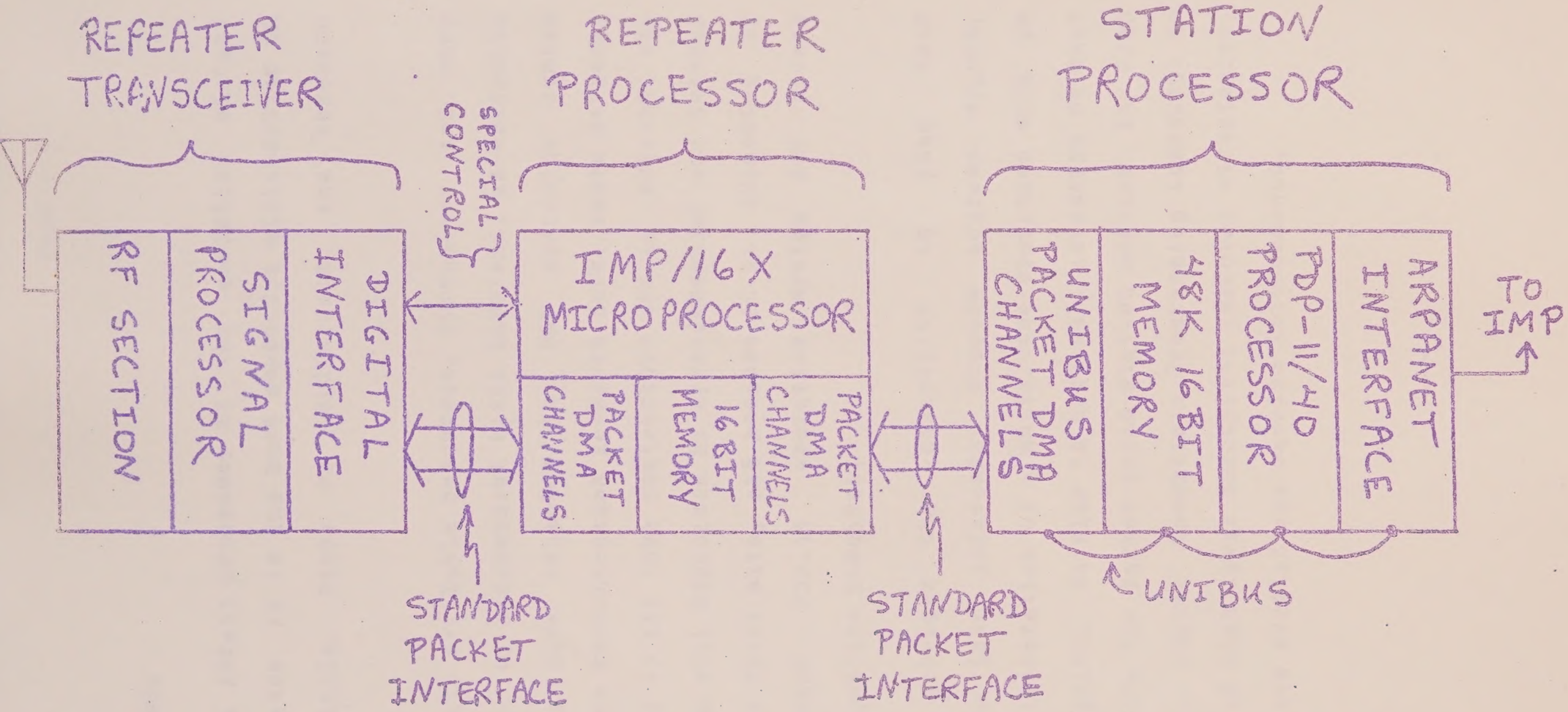


Figure 1: Structure of Packet Radio Station

RF SECTION
PROCESSOR
SIGNAL
INTERFACE
DIGITAL
TRANSCEIVER
CONTROL
STATUS
HOST
INTERFACE
PACKET
RAM
CHANNELS
MEMORY
16 BIT
PACKET
RAM
CHANNELS
MEMORY
SEQUENCER
PROCESSOR
HOST
INTERFACE
PACKET RAM
CHANNELS
MEMORY
16 BIT
PROCESSOR
DSP
SEQUENCER
PROCESSOR
ARGUMENT
HOST
INTERFACE

A. The Station Processor

We have selected a PDP-11 processor as the computing element for the station because it is the best supported minicomputer in terms of software available, range of peripherals, and hardware maintenance.

In order to allocate available resources, (processor and core memory) to the number of independant tasks involved, an operating system is required. We have selected the ELF operating system developed by the Speech Communication Research Labs, Santa Barbara, because it is available to all ARPA contractors and in current use by many, because it is well supported by crossassembler and network bootstrap loader at any TENEX site, and because it is supported by ARPA as its standard PDP-11 operating system for Speech Understanding and Recognition research.

The ELF system provides the capability to load user applications programs into independnt process address spaces, permitting modular implementation of a multi-task operation such as the Packet Radio Station. We plan to take full advantage of that capability by programming the Packet Radio Station modules in high level language, BCPL, (Basic Compatible Programming Language) to obtain the advantages of high level languages; improved structuring, simplified debugging, and substantial machine independence.

Of course, even though ELF provides a suitable environment for execution of such applications modules, it does not provide a very supportive programming environment. To remedy this, we plan to do all programming on TENEX systems, making use of a BCPL cross-compiler to generate PDP-11 code, a network bootstrap loader to transfer the resulting core image into the PDP-11, a cross-network BCPL debugger which keeps all symbol trees, command and instruction interpreters in TENEX. In addition, TENEX's file system, editors, and other utilities will greatly simplify the programming development effort. The new facilities which must be integrated into ELF are minimal: a small (about 128 word) network bootstrap loader, and a mini-debugger monitor (of similar size) which is capable of doing examine, deposit, start and stop as directed by network messages from a TENEX system.

The Station Processor will include a full duplex packet DMA (direct memory access) interface which provides all the signals specified in Specification of a Standard Packet Interface. This unit will be constructed by BBN using two DEC standard DR-11B memory channels to provide full duplex access to the Unibus and PDP-11 memory. Also included is a special host interface to the ARPA network.

B. The Repeater Processor

The Repeater Processor is the same module used in the repeater, the station, and the terminals. It consists of a special

Digitized by the Internet Archive
in 2025 with funding from
Amateur Radio Digital Communications, Grant 151

https://archive.org/details/temporary-note-105

modification (designed by Collins Radio) of the National Semiconductor 16-bit microprocessor, called the IMP16/X. It includes 1K words of random access 16 bit memory, expandable to 8K words for the terminal application. Full-duplex direct memory access channels are provided which conform to the Specification of a Standard Packet Interface. One such interface is used in the repeater, terminal, and station for interconnection with the Repeater Transceiver, and one additional instance of this interface is used in the station for interconnection with the Station Processor.

C. The Repeater Transceiver

The Repeater Transceiver is used in the repeater, the terminals, and the stations; it contains an RF receiver and transmitter, a signal processor which supports the spread-spectrum communications, and a digital interface which performs error checksumming and serial/parallel conversion. The digital interface supports full duplex data movement with a Standard Packet Interface, and in addition has a number of direct control/status connections to the IMP16/X microprocessor permitting flexible program control of the transceiver operation.

D. Observations

The structure described above has the following advantages:

1. Standard hardware modules are used in common by the repeater, the terminal, and the station.

2. The use of the Standard Packet Interface permits easy self-looping and cross-coupling experiments. For example, a planned experiment will directly cross-couple the Repeater Processors in a station and a terminal. This will permit early experiments in station-terminal protocols without any complications due to RF propagation or interference.

XII. Functional Partitioning

It is not only desirable to use standard hardware in common for the station, repeater, and terminal; we would also like to maximize the commonality of software in the three. The functional partitioning described below attempts to achieve this.

A. Station Communication Module: The Level 9 Repeater

The Station Communication Module contains the hardware of a repeater; it should also contain the software. This is a natural result if we characterize the station as a Level 9 Repeater. The station's repeater microprocessor is responsible for forwarding messages both upstream (toward the station processor) and downstream (to other repeaters). It is responsible for receiving and sending the hop-by-hop acknowledgements. It is also responsible for initiating and responding to searches in the same way other repeaters do.

The only differences between the station's repeater and other repeaters in the network are the following:

1. Upstream communications (to the station processor) go through a special channel rather than through the transceiver.

2. The protocol used on this channel is extremely simple: messages are transferred without the need for hop acknowledgements or searches. (This interconnection is assumed quite reliable compared to a radio link).

3. The last hop acknowledgement for incoming traffic must be generated and output through the transceiver.

It is worth noting that three similar considerations apply to the terminal, so that a good deal of microprocessor software can be used in common between the station and the terminal, in addition to the repeater software which is common to all packet radio nodes.

Using this partitioning to maximize commonality of software, the functions which belong in the station processor are the following:

1. End-to-end flow control and error control using an (as yet unspecified) station-terminal protocol.

2. Global routing decisions; labelling of repeaters and tracking of terminals.

3. Gateway functions to convert between protocols of the ARPANET and the Packet Radio Net.

4. Terminal control; establishing TELNET connections through the ARPANET and implementing the parts of the TELNET protocol which cannot reasonably be implemented in each PRN terminal.

Detailed examination of the four areas above are beyond the scope of this note.

B. The Superstation

To guarantee continued operation of the Packet Radio Network in spite of component failures, it is necessary to have more than one station in each radio area. In addition, it must be possible to pass control of a terminal from one station to another as the terminal moves through this radio area. To facilitate this process, we introduce the idea of a superstation to form Level -1 of the Packet Radio Network. (See Figure 2).

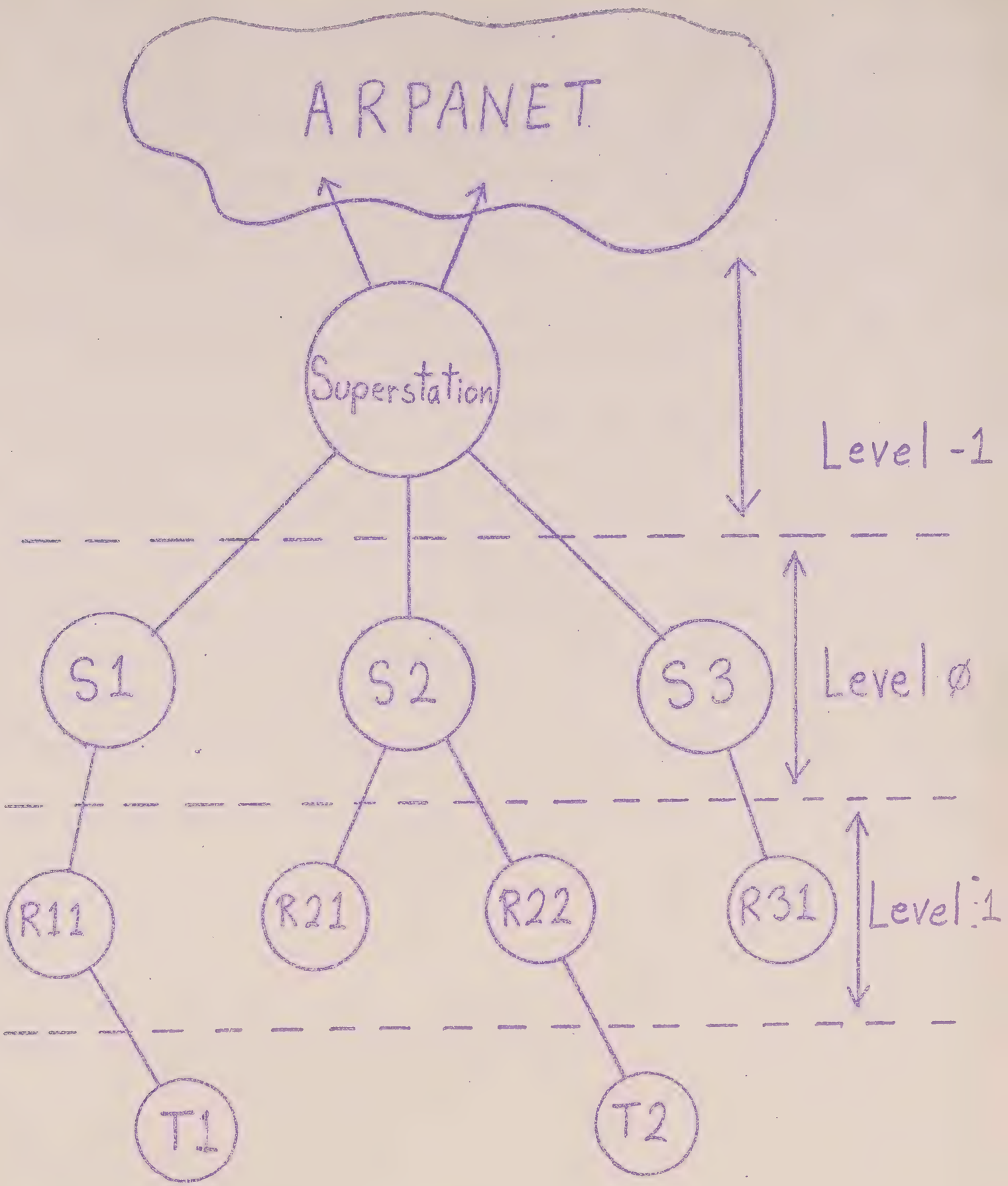


Figure 2: Logical Structure of Radio Coverage Area

All the stations in the radio coverage area are conceptually Level Ø repeaters, which communicate upstream with a single virtual entity at Level 1, known as the Superstation. If any stations (Level Ø repeaters) go out of operation, the standard PRN search algorithms will find a route through some other station to the Superstation, and communication can continue normally.

For reliability, the Superstation cannot be implemented on a single machine. Instead, it is a distributed operating system executing simultaneously on all station processors in the radio coverage area. Its database exists as redundant copies at all these stations, which are updated whenever there is a significant change. When a station fails, its duties are picked up by any other accessible station, which will have a database adequate to resynchronize and resume communications. (See Figure 3).

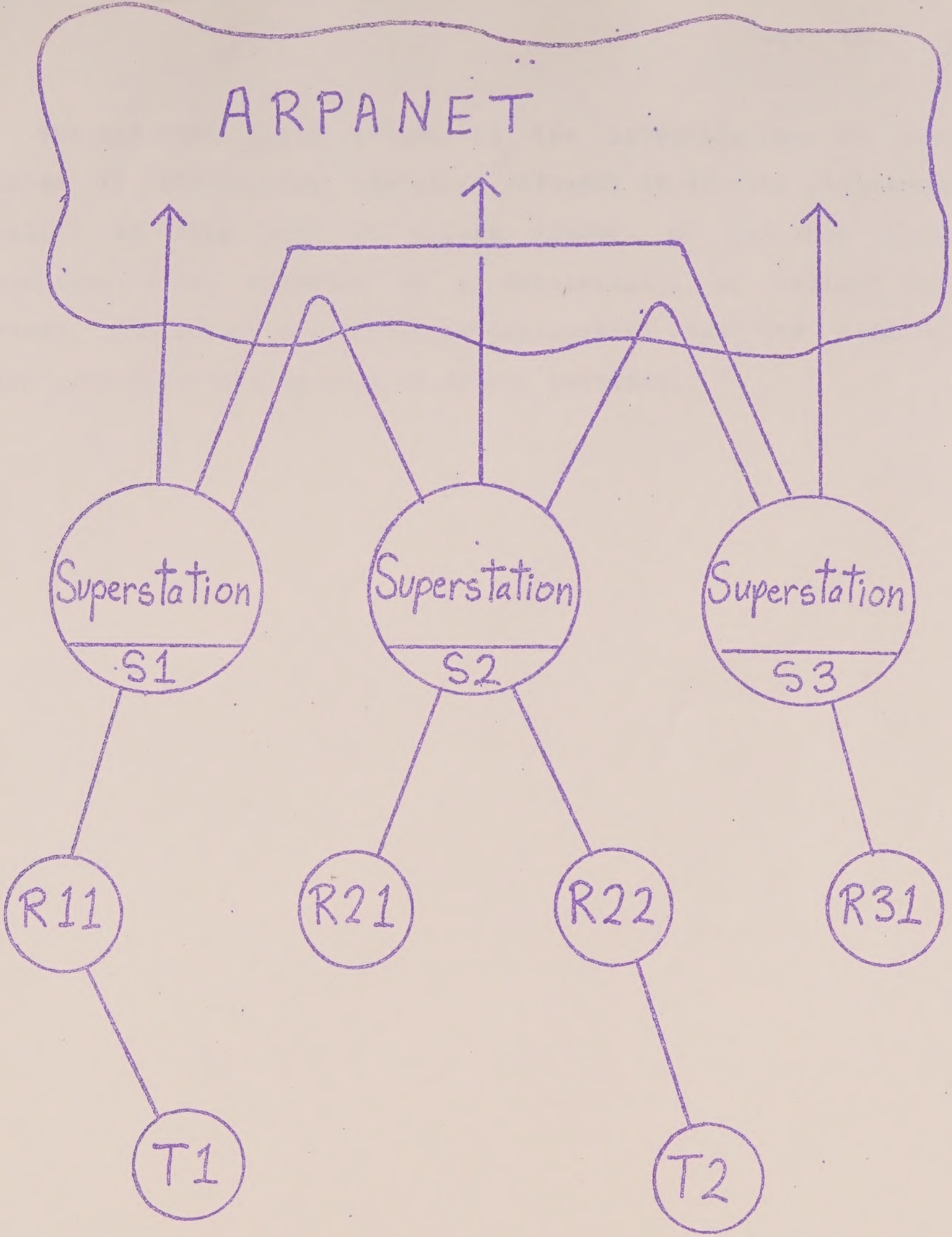
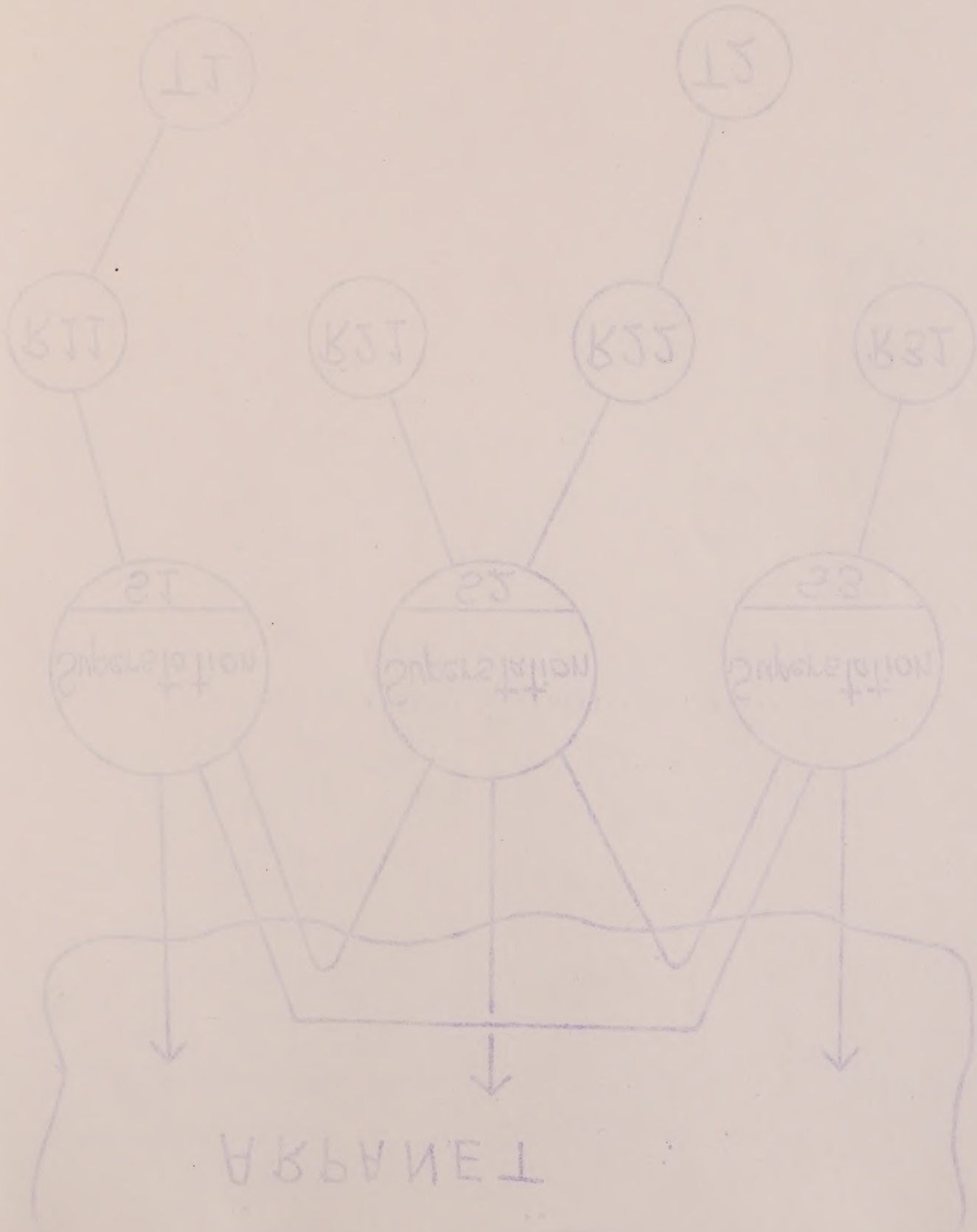


Figure 3: Implementation of Superstation


ARPANET

BBN has been quite active in the investigation of issues related to distributed operating systems; an example experimental facility of this sort is called RSEXEC, or Resource Sharing Executive, which executes on a multiplicity of ARPANET TENEX systems. The Superstation will be implemented using the techniques which were developed as part of RSEXEC research.